

# Verifying safety of an autonomous spacecraft rendezvous mission (Experience Report)

Nicole Chan and Sayan Mitra

Coordinated Science Laboratory,  
University of Illinois at Urbana-Champaign  
{nschan3,mitras}@illinois.edu

## Abstract

A fundamental maneuver in autonomous space operations is known as rendezvous, where a spacecraft navigates to and approaches another spacecraft. In this case study, we present linear and nonlinear benchmark models of an active chaser spacecraft performing rendezvous toward a passive, orbiting target. The system is modeled as a hybrid automaton, where the chaser must adhere to different sets of constraints in each discrete mode. A switched LQR controller is designed accordingly to meet this collection of physical and geometric safety constraints, while maintaining liveness in navigating toward the target spacecraft. We extend this benchmark problem to check for passive safety, which is collision avoidance along a passive, propulsion-free trajectory that may be followed in the event of system failures. We show that existing hybrid verification tools like SpaceEx, C2E2, and our own implementation of a simulation-driven verification tool can robustly verify this system with respect to the requirements, and a variety of relevant initial conditions.

## 1 Introduction

A new age of deep space exploration is underway with several ongoing public-private partnerships. A groundwork for a possible mission to Mars in the 2030s is also underway [16]. Autonomous operations where a spacecraft can operate independent of human control in a wide variety of conditions are essential for deployment, construction, and maintenance missions in space. Despite many spectacular successes like the Mars landing of the Curiosity rover, ensuring safety of autonomous spacecraft operations remains a daunting challenge. The cost of failures can be extreme. For example, NASA’s DART spacecraft was designed to rendezvous with the MUBLCOM satellite. In 2005, approximately 11 hours into a 24-hour mission, DART’s propellant supply depleted due to excessive use of thrusters, and it began maneuvers for departure. In the process it collided with MUBLCOM; it met only 11 of 27 mission objectives, and the failure resulted in a loss exceeding \$1 million. In another incident, a navigation error caused the Mars Climate Orbiter to reach as low as 57 kilometers, where it was intended to enter an orbit at an altitude of 140-150 kilometers. The spacecraft was destroyed by the resulting atmospheric stresses and friction and the cost incurred was \$85 million. These incidents, and several others in [21] highlight the consequences of failures in space applications and demonstrate the need for more rigorous testing before deployment.

Although formal verification has played an important role in design and safety analysis of spacecraft hardware and software (see, for example [10] and the references therein), they have not been used for model-based design and system-level verification and validation. In this paper, we present and verify a realistic and challenging spacecraft maneuver problem called autonomous rendezvous, proximity operations, and docking (ARPOD). The original hybrid control design problem and its variants are introduced by Jewison and Erwin in [13]. ARPOD is a fundamental set of operations for a variety of space missions such as on-orbit transfer of personnel [20], resupply for on-orbit personnel [17], assembly [22], servicing, repair, and refueling [9].

The basic setup for ARPOD consists of a passive module or a *target* (launched separately into orbit) and a *chaser* spacecraft that must transport the passive module to an on-orbit assembly location. The chaser maintains a relative bearing measurement to the target, but initially it may be too far away from the target to use its range sensors. Range measurements become available within a given range, giving the chaser accurate relative positioning data so that it can stage itself to dock the target. The target must be docked with a specific angle of approach and closing velocity, so as to avoid collision and ensure that the docking mechanisms on each body will mate. Furthermore, the docking procedure must be completed before the chaser goes into eclipse and loses vision-based sensor data. Finally, it is necessary for the system to ensure *passive safety*. That is, the chaser spacecraft should maintain safe separation from the target even if it loses power and communication during its mission.

In this paper, we present a suite of hybrid models for the rendezvous portion of the ARPOD mission that can serve as benchmarks for verification tools and serve as building-blocks for more complex operations. We present nonlinear models that consist of nonlinear orbital dynamics (NLinProx and NLinProxTh) and linearized models (LinProx and LinProxTh) using the Clohessy-Wiltshire-Hill (CWH) equations [1]. The rendezvous operation is further subdivided into two phases: Proximity Operations A and B, such that phase A captures an interval of larger ranges than the interval of ranges in phase B. In other words, the chaser enters phase A first and as it moves closer to the target, it enters phase B. After this two-phase rendezvous, the chaser enters a docking phase/maneuver (i.e. when the chaser is less than 10 meters from its target). We disregard the requirements for this phase for this paper. We develop a switched state-feedback controller using Linear Quadratic Regulation (LQR) for the rendezvous phases. The position-triggered transitions brought about by the switching controller are urgent, resulting in a deterministic hybrid automaton. However, we extend the ARPOD problem in [13] to include the passive safety requirement by introducing a nondeterministic time-triggered transition to the passive mode.

We have successfully verified the requirements for most of the models using existing hybrid verification tools SpaceEx [8] and C2E2 [3, 5], and also our new MatLab implementation of a simulation-driven verification algorithm (SDVTool) for linear hybrid models. SDVTool improves C2E2's reachability algorithm, with a new technique [4] for obtaining reachsets for linear systems. We obtain verification results for an array of varied initial state configurations and passive transitions times to show the robustness and limits of the switched LQR controller. The experiments, in particular on the passive safety requirement, have demonstrated a weakness in simulation-driven verification approaches in handling ill-conditioned models which suggest a need for further research. Overall, we believe that our results and approaches establish feasibility of system-level verification of autonomous space operations, and they provide a foundation for the analysis of more sophisticated maneuvers in the future.

## 2 Related work

There are few academic works on system-level verification of autonomous spacecraft. A survey of general verification approaches and how they may apply to small satellite systems is presented in [12]. Architecture and Analysis Design Language (AADL) and verification and validation (V&V) over AADL models for satellite systems have been reported in [2]

An feasibility study for applying formal verification of autonomous satellite maneuvers is presented in [15]. That approach relied on creating rectangular abstractions (dynamics of the form  $\dot{x} \in [a, b]$ ) of the satellites dynamics through hybridization and verification using PHAVer [7] and SpaceEx [8]. The generated abstract models have simple dynamics but hundreds of locations,

and also, the analysis is necessarily conservative. In contrast, the approaches presented in this paper work directly with the linear (nonlinear) hybrid dynamics.

The ARPOD challenge [13] has been taken up by several researchers in proposing a variety of control strategies. A two-stage optimal control strategy is developed in [6], where the first part involves trajectory planning under a differentially-flat system and the second part implements Model Predictive Control on a linearized model. A supervisor is introduced to robustly coordinate a family of hybrid controllers in [18]. Safe reachsets (i.e. ReachAvoid sets) are computed for the ARPOD mission in [11] and used to solve for minimum fuel and minimum time trajectories.

### 3 Spacecraft Rendezvous Model

In this section, we present the detailed development of the hybrid models. First we present the orbital dynamics of the spacecraft in Sections 3.1-3.2. Then in Sections 3.3-3.4 we present a hybrid controller. Finally, we state the various mission constraints in Section 3.5.

#### 3.1 Nonlinear relative motion dynamics

The dynamics of the two spacecraft in orbit—the *target* and the *chaser*—are derived from Kepler’s laws. We use the simplest case for relative motion in space, where the two spacecraft are restricted to the same orbital plane, resulting in two-dimensional, planar motion. The so called Hill’s relative coordinate frame is used. As shown in Figure 2, Hill’s frame is centered on the target spacecraft, with  $+\hat{\mathbf{i}}$ -direction pointing radially outward from the Earth,  $+\hat{\mathbf{k}}$ -direction normal from the orbital plane, and  $+\hat{\mathbf{j}}$ -direction completing a right-handed system. We further assume that the target moves on a circular orbit, and thus, the  $\hat{\mathbf{j}}$ -direction aligns with the tangential velocity of the target.

The restriction on the target’s orbit implies that the target-centered frame rotates with constant angular velocity. We will assume the target is in geostationary equatorial orbit (GEO), so its angular velocity is  $n = \sqrt{\frac{\mu}{r^3}}$ , where  $\mu = 3.698 \times 10^{14} m^3/s^2$  and  $r = 42164 km$ . The chaser’s position is represented by the vector  $x\hat{\mathbf{i}} + y\hat{\mathbf{j}}$ , and the chaser’s thrusters provide acceleration in the form of  $F_x\hat{\mathbf{i}} + F_y\hat{\mathbf{j}}$ . The following equations are derived using Kepler’s laws and constitute the nonlinear model of the spacecraft dynamics.

$$\begin{aligned}\ddot{x} &= n^2x + 2n\dot{y} + \frac{\mu}{r^2} - \frac{\mu}{r_c^3}(r+x) + \frac{F_x}{m_c}, \\ \ddot{y} &= n^2y - 2n\dot{x} - \frac{\mu}{r_c^3}y + \frac{F_y}{m_c},\end{aligned}\tag{1}$$

where  $r_c = \sqrt{(r+x)^2 + y^2}$  is the distance between the chaser and Earth and  $m_c = 500kg$  is the mass of the chaser.

#### 3.2 Linear dynamics

Linearization of these equations about the system’s equilibrium point results in the Clohessy-Wiltshire-Hill (CWH) equations [1], which are commonly used to capture the relative motion

dynamics of two satellites within a reasonably close range. These equations are:

$$\begin{aligned}\ddot{x} &= 3n^2x + 2n\dot{y} + \frac{F_x}{m_c}, \\ \ddot{y} &= -2n\dot{x} + \frac{F_y}{m_c}.\end{aligned}\tag{2}$$

Let the state vector be denoted by  $\vec{x} = [x, y, \dot{x}, \dot{y}]^T$ . The state-space form of these linear time-invariant (LTI) equations is:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}, \text{ where,}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix}, \vec{u} = \begin{bmatrix} F_x \\ F_y \end{bmatrix}.$$

### 3.3 Hybrid controller model

Complete hybrid automaton models for the system with additional documentation are available from [from this link](#)<sup>1</sup>. Varying ranges of the relative distance between the spacecraft give rise to different constraints and requirements, and therefore, require separate controllers. We present a two-stage hybrid controller for achieving the rendezvous maneuver<sup>2</sup>. We refer to these discrete stages as *modes*. Each discrete mode has an *invariant* which specifies the conditions under which the system may operate in that mode, which we will first describe in words.

*Mode 1* or Proximity Operations A (ProxA): the chaser is attempting to rendezvous and its separation distance ( $\rho = \sqrt{x^2 + y^2}$ ) from the target is in the range 100-1000m.

*Mode 2* or Proximity Operations B (ProxB): the chaser is attempting to rendezvous and its separation distance is less than 100m.

*Mode 3* or Passive mode: the chaser is no longer attempting to rendezvous and is not using its thrusters, regardless of its separation distance. The system may transition to the Passive mode as a result of a failure or loss of power.

The state of the overall hybrid system is defined by the mode and the valuations of a set of continuous variables: relative position  $x, y$ , thrusts  $F_x, F_y$ , and a global timer *clock*. There are two timing parameters of the model  $t_1$  and  $t_2$  that specify the time interval over which the chaser spacecraft may enter the Passive mode. When the system is in a particular mode, the continuous variables  $(x, y)$  evolve according to the (linear or nonlinear) differential equations of the previous section. The thrust inputs  $F_x$  and  $F_y$  are computed according the full-state feedback controller designed in Section 3.4.

We refer to the time elapsed in the mission with the variable *clock* but do not consider it an explicit state variable. The invariants in each mode can be more precisely described as  $\rho \geq 100$  and  $clock \leq t_2$  for mode 1,  $\rho \leq 100$  and  $clock \leq t_2$  for mode 2, and  $clock \geq t_1$  for mode 3. A transition from one mode to another is described by a *guard*. When the state satisfies the guard condition, the system *may* take the transition. If a transition is required to occur as soon as possible, this is called an *urgent transition*. In this system, the distance-based

<sup>1</sup><https://tinyurl.com/verifysat>

<sup>2</sup>The rendezvous mission presented in this paper is a subset of the four-stage problem presented in [13]. Our two stages of rendezvous are almost identical to “Phase 2” and “Phase 3” in [13].

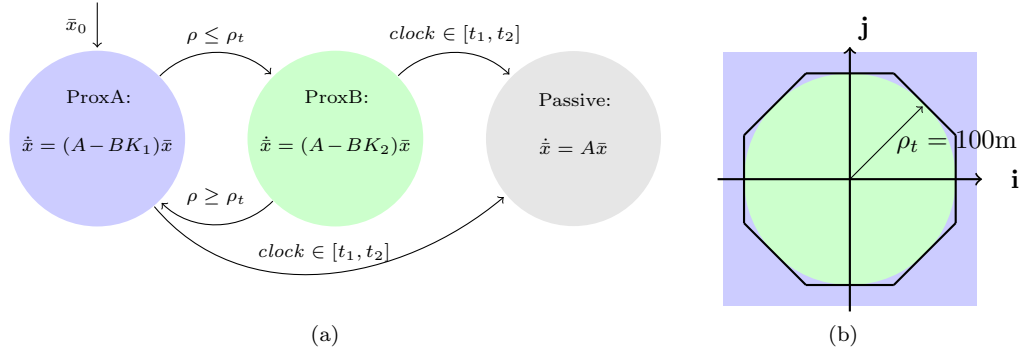


Figure 1: (a) Hybrid model for spacecraft rendezvous, with linear flow equations shown. The invariants in ProxA and ProxB are defined exclusively by the chaser's position, as shown by corresponding colors in the plane of motion in (b). The transition guards between ProxA and ProxB align exactly with their invariant sets, resulting in urgent transitions. The invariant for Passive mode is  $clock > t_1$ , irrespective of position. A transition to Passive occurs sometime within an interval of time, and hence is nondeterministic. In (b), the octagon represents how the invariants/guards are approximated and modeled in the verification tools.

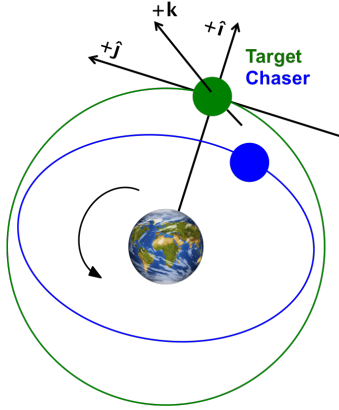


Figure 2: Hill's relative coordinate frame. The chaser's relative position vector is  $x\hat{i} + y\hat{j}$ .

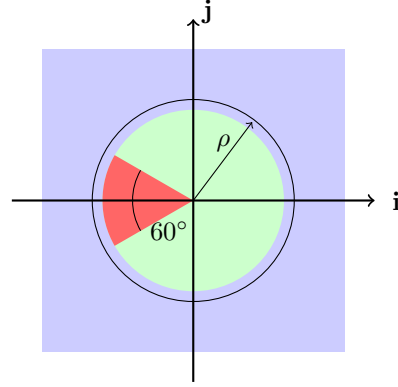


Figure 3: The hybrid model (see Figure 1a) captures the chaser's motion in ProxA (blue) and ProxB (green), and we use verification tools to show that whenever  $\rho \in \text{ProxB}$ , the chaser does not leave the LOS region (red).

transitions between modes 1 and 2 are urgent. However, the transitions to mode 3 (Passive mode) are not urgent. There is an interval of time,  $clock \in [t_1, t_2]$ , within which the chaser could nondeterministically transition to the Passive mode. Roughly, a larger  $[t_1, t_2]$  interval implies a bigger passive-safety envelope for the mission. These transitions to the Passive mode make the system nondeterministic. Indeed, for some choices of this interval, it is possible for the hybrid system to occupy any one of the three modes at a given time.

### 3.4 Linear Quadratic Control

We have developed a full-state feedback controller, namely, a Linear Quadratic Regulator (LQR), to drive the chaser towards the target's position. Closed-loop feedback is desirable because the system can measure and adjust for errors, and ultimately guarantee liveness (i.e. eventually the target will be reached). LQR is specifically chosen because it is constructed by minimizing a quadratic cost function, which we can choose so as to roughly satisfy our safety constraints. LQR is only applicable to linear systems, so we design the control for the linearized model in (2), but we will use the same control with nonlinear dynamics (1) when applying verification tools.

The form of an LQR solution is:  $\vec{u} = -K\vec{x}$ , where  $K \in \mathbb{R}^{4 \times 2}$  is a constant matrix and  $\vec{u} = [\frac{F_x}{m_c}, \frac{F_y}{m_c}]^T$ . The  $K$  matrix is found by minimizing the following cost function with respect to  $\vec{u}$ :  $\int_0^\infty (\vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u}) dt$ , where  $Q$  and  $R$  are positive definite matrices.

Given the form of the control law, we update the definition of the model given in (2) to the following:

$$\begin{aligned}\ddot{x} &= \left(3n^2 - \frac{k_{11}}{m_c}\right)x - \frac{k_{12}}{m_c}y - \frac{k_{13}}{m_c}\dot{x} + \left(2n - \frac{k_{14}}{m_c}\right)\dot{y}, \\ \ddot{y} &= -\frac{k_{21}}{m_c}x - \frac{k_{22}}{m_c}y - \left(2n + \frac{k_{23}}{m_c}\right)\dot{x} - \frac{k_{24}}{m_c}\dot{y}.\end{aligned}\tag{3}$$

These equations are the expanded version of the closed-loop form shown in Figure 1a. Later, we will discuss why we distinguish between the dependence of (2) on  $\vec{x}$  and  $\vec{u}$  and (3) on only  $\vec{x}$ .

Bryson's method [14] is used to help determine an appropriate cost function. Begin with  $Q$  and  $R$  as diagonal matrices, and choose their values so as to normalize each of the state and input variables. In other words, choose the diagonal elements so that  $Q_{ii} = \frac{1}{\max(x_i^2)}$  and  $R_{ii} = \frac{1}{\max(u_i^2)}$ . Here, the denominators refer to the largest *desired* value of each variable, which will be determined by the safety constraints and mode invariants. While the LQR gains are obtained with our constraints in mind, the resulting controller does not guarantee these constraints are never violated. This is why further verification is still required. This design process is repeated for modes 1 and 2, and the result is two distinct LQR controllers for each of these modes in our hybrid system.

### 3.5 Constraints and safety requirements

In this section, we enumerate the properties that define a safe and successful mission, and how they are modeled for verification tools.

**Thrust constraints** During the rendezvous stages (ProxA and ProxB), the thrusters cannot provide more than  $10N$  of force in any single direction, therefore, we have the constraints:

$$|F_x|, |F_y| \leq 10.$$

**LOS cone and proximity** During close-range rendezvous ProxB, the chaser must remain within a line-of-sight (LOS) cone (see Figure 3), and its total velocity must remain under 5cm/s, so  $\sqrt{\dot{x}^2 + \dot{y}^2} \leq 5\text{cm/s}$ . The total velocity constraints cannot be exactly modeled using linear constraints, and a polytopic approximation over  $\dot{x}, \dot{y}$  is used. This is done in the same way as  $\sqrt{x^2 + y^2} \leq \rho_t$  is approximated (see Figure 1b).

**Separation** During the Passive mode, the chaser must avoid collision with the target, which is theoretically a point mass at the origin. Even in a theoretical model, a small ball or box should be used to bound this point to account for limitations in numerical precision. In reality, the target satellite’s dimensions may range from the order of 1m to 100m, so the size of this bounding box will vary depending on the situation. We use a box with a 0.1m circumradius.

## 4 Verification approaches

In this section, we briefly discuss our experience in using hybrid system verification tools. **SpaceEx** [8], is a well-established reachability analysis tool for linear and affine hybrid systems. It implements the support function-based reachability algorithm, includes the PHAVer algorithm for rectangular dynamics [7], and also a simulator for nonlinear models. The support function representation of sets is amenable to effective computation of convex hulls, linear transforms, Minkowski sums, etc.—operations that are necessary for safety verification.

**C2E2** [3, 5] is a simulation-driven bounded verification tool for nonlinear hybrid models. The core algorithm of C2E2 relies on computing reachset over-approximations from validated numerical simulations and what are called *discrepancy functions*. A discrepancy function for a model bounds the sensitivity of the trajectories of the hybrid system to changes in initial states and inputs. Candidate discrepancy functions can be obtained using a global Lipschitz or using a matrix norm for linear systems. However, typically these approaches give discrepancy functions that blow-up exponentially with time, and therefore, are not useful for verifying problems with long time horizons. The automatic on-the-fly approach implemented in [5] uses bounds on the Jacobian matrix of the system to get tighter local discrepancy functions and it has been used to verify several benchmark problems. Recently the tool has been extended to handle nonlinear models with dynamics with exponential and trigonometric functions.

For a (possibly nonlinear) mode with  $\dot{x} = f(x(t))$ , the discrepancy computed by the algorithm of [5] uses the Jacobian matrix  $J(x)$  of  $f(x)$  and the condition number of  $J(x_0)$  evaluated at certain points  $x_0$  in the state space. For ill-conditioned matrices, such as what we have in the passive mode, (the  $A$ -matrix representation of (2)), the over-approximation error may still blow-up. Ill-conditioned systems may not only arise from passive dynamics but also from extremely large and small coefficients appearing together in  $J(x_0)$ .

In order to address this problem, we have created a MATLAB implementation of C2E2’s verification algorithm (**SDVTool**) for linear models. Unlike C2E2, **SDVTool** does not rely on discrepancy, but instead computes the reachable states under a given linear mode directly. The particular algorithm implemented is the one presented in [4]: For an  $n$ -dimensional system,  $n + 1$  simulations are performed. From these simulations, special sets called *generalized star sets*, are generated to represent the exact reachsets. For our purposes, a generalized star set is represented by a pair  $\langle x_0, V \rangle$ , where  $x_0 \in \mathbb{R}^n$  is the center state and  $V = \{v_1, \dots, v_n\} \subseteq \mathbb{R}^n$  is a standard basis (not necessarily unit vectors), and the set defined by  $\langle x_0, V \rangle$  is

$$\{x \in \mathbb{R}^n \mid \exists \alpha_1, \dots, \alpha_n \in [-1, 1], x = x_0 + \sum_{i=1}^n \alpha_i v_i\}.$$

As reachsets are calculated for time steps,  $x_0$  and  $V$  are transformed. When the reachtube from a given mode intersects the guards for a transition, the star sets are aggregated and over-approximated with hyperrectangles. If  $R_i^* = \langle x(t_i), V_i \rangle$  is the star set reachset obtained



at time  $t_i$ , then the hyperrectangular reachset is:

$$R_i = \{x \mid x \leq x(t_i) + \sum_{j=1}^n \max(-v_j, v_j) \\ \text{and } x \geq x(t_i) + \sum_{j=1}^n \min(-v_j, v_j)\}.$$

C2E2 and SDVTool currently accumulates all the reachable sets in ProxA and ProxB that *may* transition to Passive, and uses their convex hull to begin reachset computations under the Passive mode. It follows that if the time interval during which a transition may occur is large, then the initial set of states under the Passive mode is large, making it very difficult to prove safety. One solution is to allow partitioning and refinement of the initial passive mode set. Since this is not currently implemented in C2E2 or SDVTool, we restrict our experiments to transition interval lengths of 5 minutes or less. For example, checking if the system is safe for a transition  $clock \in [50, 200 \text{ min}]$  could be achieved by running several experiments with small subintervals that cover the original interval.

## 5 Verification results

In this section, we discuss and compare verification results from SpaceEx, C2E2, and our implementation of SDVTool. Based on these results, we reach the broad conclusion that with some manual tweaks, the current hybrid system verification tools are indeed capable of analyzing realistic system-level properties of autonomous spacecraft maneuvers.

In the following presentation, we pick arbitrary model parameters, but to a large extent our results are robust with respect to parameter variations. That is, the parameters can be tuned to the specific requirements of a real mission.

For subsequent discussion, we label our models as follows: LinProx denotes the equations in (3), NLinProx denotes the equations in (1) with the same controller as LinProx substituted into  $F_x, F_y$ , and LinProxTh denotes a model that will soon be introduced to account for explicit thrust values.

### 5.1 Hybrid safety proofs

Figure 4 shows the typical reachset computations obtained from SDVTool, C2E2, and SpaceEx on the LinProx, LinProxTh, and NLinProx models. These computations also establish the safety of the corresponding systems with respect to the requirements in Section 3.5. Overall, the plots show that the reachsets from the different tools are qualitatively similar. From the more detailed MatLab plots we can check that no part of the reachable sets intersect with unsafe regions. It is clear from the zoomed in portion of Figure 4a that a reasonably larger collision region would violate safety.

In C2E2 and SpaceEx, each safety property is loaded and checked individually. In C2E2, the running time for a single property for the nonlinear model NLinProx is in the neighborhood of 5-10 minutes; in SpaceEx, the running time for a single property is on the order of a few seconds. SDVTool checks all (12) properties simultaneously and the running time varies from around 30 seconds to 10 minutes. We do not compare absolute running times in further detail in this paper as each of the tools have different semi-automatic workflows and require widely different execution environments.



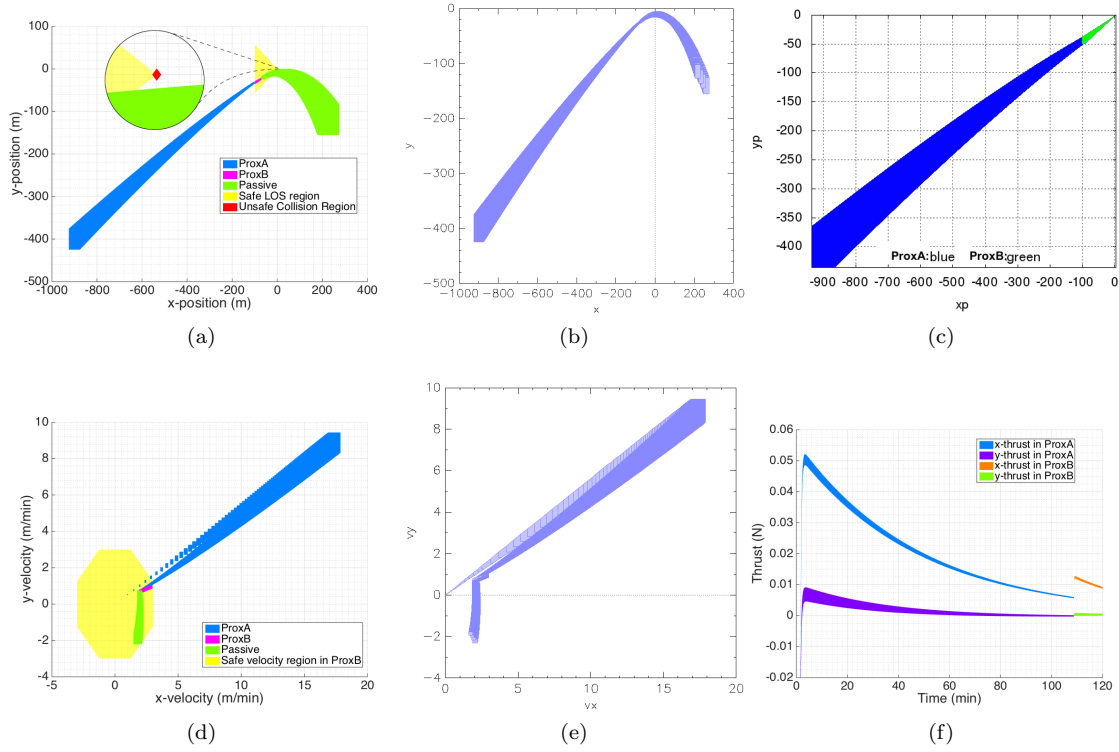


Figure 4: Examples of various generated reachsets. Reachable positions using LinProx in SDVTool shown in (a) and SpaceEx in (b). Reachable velocities using LinProx in SDVTool shown in (d) and SpaceEx in (e). Reachable positions of NLinProx without Passive mode in C2E2 shown in (c). Reachable thrusts using LinProxTh in SDVTool is shown in (f).

**Time horizon** Timing is obviously critical for space applications to ensure there is sufficient fuel, but with over-approximated reachability, we can only guarantee the mission is completed within a time upper-bound. This upper-bound obtained from reachability analysis may differ significantly from the actual mission time. Therefore, we do not impose any strict completion requirements. Instead, we choose a time horizon that is representative of what we might expect in practice, and focus on observing what behaviors are possible within these limits. Typically, Proximity A operations take 1-5 orbits (at under 4 hours an orbit) and proximity B operations take 45-90 minutes [19]. We choose a sum of approximately 4.5 hours to be our time horizon.

**Initial states** We calculate a set of initial states assuming that the chaser spacecraft is performing the encompassing mission from [13]. We choose an initial set radius of  $[25m, 25m, 0, 0]$  around the point  $\vec{x}_0 = [-900m, -400m, 0m/s, 0m/s]^T$ . This can be interpreted as uncertainty in the chaser's initial position, typically due to loss of precision from sensors and computations, or it can be used to explore multiple initial states of interest. We have successfully verified scenarios with uncertainty in the velocity dimensions as well.

**Unsafe sets** For SpaceX, C2E2, and SDVTool, we model the safety requirements as a collection of linear inequalities. The LOS cone is approximated with a triangle, so we check three properties to prove the system remains within LOS constraints, and so on. Max thrust is effectively a one-dimensional constraint, a nonconvex interval, so two properties will capture the unsafe set for each thrust input (one along  $x$ -direction, one along  $y$ -direction). But in order to treat it this way, we must introduce extra variables  $u_x, u_y$  to explicitly track the thruster values. These extra variables are the difference between LinProx and LinProxTh.

**Passive transition time** The interval of time during which a transition to the passive mode may occur is trivially bounded by the mission time horizon. For this first example, we choose a small interval at [120, 125min]. This ensures that the chaser will operate in mode ProxB before transitioning to the passive mode.

## 5.2 Adding thrust constraints

In Section 3.5, we described a constraint on thrust that mimics the physical limitations of our spacecraft. We now set up the 6-dimensional model LinProxTh so that we can verify this additional requirement. We introduce  $u_x, u_y$  as explicit state variables, and solve for their differential equations to obtain:

$$\begin{aligned}\dot{u}_x &= k_{11}\dot{x} + k_{12}\dot{y} + k_{13}\ddot{x} + k_{14}\ddot{y}, \\ \dot{u}_y &= k_{21}\dot{x} + k_{22}\dot{y} + k_{23}\ddot{x} + k_{24}\ddot{y}.\end{aligned}\tag{4}$$

There are two equivalent numerical models that will produce different over-approximated reachsets. The first model consists of (4) and the following:

$$\begin{aligned}\ddot{x} &= 3n^2x + 2n\dot{y} - \frac{u_x}{m_c}, \\ \ddot{y} &= -2n\dot{x} - \frac{u_y}{m_c}.\end{aligned}\tag{5}$$

Here  $\ddot{x}$  and  $\ddot{y}$  account for the effects of thrust inputs by explicitly adding  $u_x, u_y$ . Since each dimension is over-approximated in the reachset computation and  $u_x, u_y$  are functions of position and velocity, the computation for subsequent reachable sets of position and velocity have even more uncertainty. Roughly,  $u_x, u_y$  act as filters for  $x, y, \dot{x}, \dot{y}$ , adding distortion and introducing more uncertainty. Figure 5 shows the effects of these compounding errors. The overarching verification algorithm will partition the initial set to reduce errors stemming from the data structure, but it will have to do this numerous times and may time out in practice.

The second 6-dimensional model (a variant on LinProxTh) consists of (3) and (4). In this case,  $\ddot{x}$  and  $\ddot{y}$  implicitly calculate the thrust, and  $u_x, u_y$  are independent “tracking” variables. The calculations for  $\ddot{x}$  and  $\ddot{y}$  are equivalent to those in (5), but they bypass the “filter” when constructing reachsets. The results are identical to those shown in Figures 4a-4d, with the addition of reachable sets of thrusts shown in Figure 4f.

Once again, our choice of data structure introduces some uncertainty to the explicit representation of the initial set of  $u_x, u_y$  values. This is propagated throughout the analysis. We use the (3)-(4) model to obtain safe thrusting results from SDVTool. The fine reachable sets in Figure 4f show that the LQR controller operates well within the thrust constraints ( $|u_x|, |u_y| \leq 10N$ ).

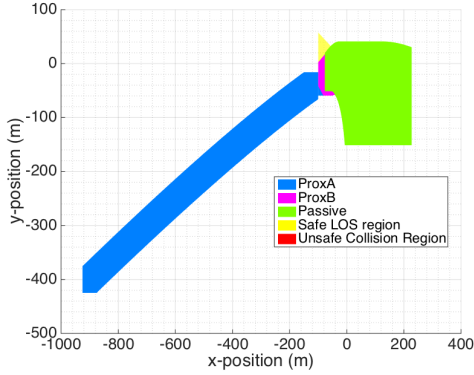


Figure 5: Coarse overapproximation of the reachable positions, when using the 6-dimensional LinProxTh model from Equation 5.

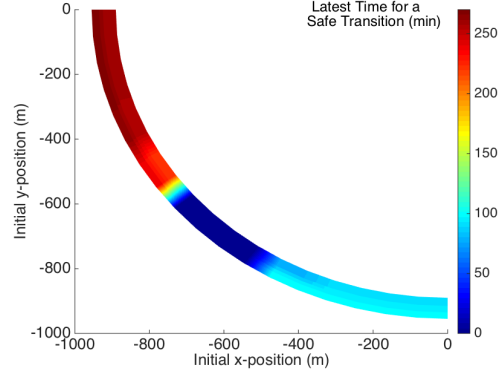


Figure 6: Initial positions (with zero initial velocities) of LinProx that have been verified to be safe. They are safe for Passive transition times up to the time shown by the color map.

### 5.3 Robustness of verification

To demonstrate the robustness of the verification approaches (and the designed controller), we performed several experiments varying the initial set and passive transition times with the LinProx model.

The scenarios that guarantee a safe mission are summarized in Figure 6. Roughly, choosing an initial position or subset of positions within the shaded region will result in a safe mission for a transition time or interval within  $[0, T]$ , where  $T$  is the time corresponding to the color at that initial position(s). For these experiments, we consider initial separation between the chaser and the target to be near  $1000m$ , where this LQR control would start being used. We assume the initial chaser velocity to be zero. Generally, we can conclude that, the closer to the  $x$ -axis the chaser starts, the later the chaser may safely abort to the passive mode. On the other hand, the neighborhood of states along  $\sim 230^\circ$  are not safe for a passive transition at any time.

## 6 Conclusions and future directions

In this case study paper, we present a sequence of linear and nonlinear, nondeterministic benchmark models of autonomous rendezvous between spacecraft with several physical and geometric safety requirements. We designed an LQR controller and verified its safety across the different models, a variety of initial conditions, parameter ranges, and using three different hybrid system verification approaches. The models and requirements are made available online.

This case study, and in particular the requirement for passive safety, has shed light on the weakness of simulation-driven verification in handling ill-conditioned models.

The results provide a foundation for verifying more sophisticated maneuvers in future autonomous space operations. For example, we proposed a continuous full-state feedback controller, but it is also possible to consider a situation where full state measurement is not possible and a simple bang-bang controller is required. Control theory tells us that this system maintains marginal stability which implies that errors will never recede, so for reasonably-sized initial sets, the reachable sets may not satisfy tight constraints such as LOS.

## Acknowledgments

We are grateful for the support of Richard S. Erwin in navigating and modeling the problem presented in this paper, and for Yu Meng's support with the C2E2 experiments. We acknowledge the support of the Air Force Research Laboratory through the Space Scholars Program.

## References

- [1] Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, sep 1960.
- [2] M. Bozzano, R. Cavada, A. Cimatti, J.-P. Katoen, V. Y. Nguyen, T. Noll, and X. Olive. Formal verification and validation of aadl models. *Proc. ERTS*, 2010.
- [3] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for stateflow models. In *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 68–82, 2015.
- [4] P. S. Duggirala and M. Viswanathan. *Parsimonious, Simulation Based Verification of Linear Systems*, pages 477–494. Springer International Publishing, Cham, 2016.
- [5] C. Fan, B. Qi, S. Mitra, M. Viswanathan, and P. S. Duggirala. Automatic reachability analysis for nonlinear hybrid models with C2E2. In *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*, pages 531–538, 2016.
- [6] S. S. Farahani, I. Papusha, C. McGhan, and R. M. Murray. Constrained autonomous satellite docking via differential flatness and model predictive control. In *CDC 2016*, Las Vegas, 2016.
- [7] G. Frehse. Phaver: Algorithmic verification of hybrid systems past hytech. In M. Morari and L. Thiele, editors, *HSCC*, volume 3414 of *Lecture Notes in Computer Science*, pages 258–273. Springer, 2005.
- [8] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. Spaceex: Scalable verification of hybrid systems. In G. Gopalakrishnan and S. Qadeer, editors, *CAV*, volume 6806 of *Lecture Notes in Computer Science*, pages 379–395. Springer, 2011.
- [9] K. Galabova, G. Bounova, O. de Weck, and D. Hastings. Architecting a family of space tugs based on orbital transfer mission scenarios. In *AIAA Space 2003 Conference & Exposition*. American Institute of Aeronautics and Astronautics (AIAA), sep 2003.
- [10] G. J. Holzmann. Mars code. *Commun. ACM*, 57(2):64–73, Feb. 2014.
- [11] B. Homchaudhuri, M. Oishi, M. Baldwin, M. Shubert, and R. S. Erwin. Computing reach-avoid sets for space vehicle docking under impulsive thrust. In *To appear in Proceedings of the Conference on Decision and Control*, 2016.
- [12] S. A. Jacklin. Survey of verification and validation techniques for small satellite software development. Space tech expo, NASA Ames Research Center, May 2015.
- [13] C. Jewson and R. S. Erwin. A spacecraft benchmark problem for hybrid control and estimation. In *CDC 2016*, Las Vegas, 2016.
- [14] M. A. Johnson and M. J. Grimble. Recent trends in linear optimal quadratic multivariable control system design. *IEE Proceedings D - Control Theory and Applications*, 134(1):53–71, January 1987.
- [15] T. T. Johnson, J. Green, S. Mitra, R. Dudley, and R. S. Erwin. Satellite rendezvous and conjunction avoidance: Case studies in verification of nonlinear hybrid systems, 2012.
- [16] B. Obama. America will take the giant leap to Mars, October 2016.
- [17] D. Pinard, S. Reynaud, P. Delpy, and S. E. Strandmoe. Accurate and autonomous navigation for the ATV. *Aerospace Science and Technology*, 11(6):490–498, sep 2007.

- [18] R. G. Sanfelice, B. Malladi, E. Butcher, and J. Wang. Robust hybrid supervisory control for rendezvous and docking of a spacecraft. In *To appear in Proceedings of the Conference on Decision and Control*, 2016.
- [19] J. R. Wertz and R. Bell. Autonomous rendezvous and docking technologies - status and prospects. Spie aerosense symposium, Apr. 2003.
- [20] D. Woffinden and D. Geller. Navigating the road to autonomous orbital rendezvous. *Journal of Spacecraft and Rockets*, 44(4):898–909, 2007.
- [21] W. E. Wong, V. Debroy, and A. Restrepo. The role of software in recent catastrophic accidents. Ieee reliability society 2009 annual technology report.
- [22] D. Zimpfer, P. Kachmar, and S. Tuohy. Autonomous rendezvous, capture and in-space assembly: past, present and future. In *Proc. AIAA Space Exploration Conference*, Jan. 2005.